

Blatt 5 - Übungsgruppe (Mo - 13:45) –  
Abgabegruppe 04

Giovanni Ngodji Djeuha, Mike Lenz, Jonas Tesfamariam,  
Bastian Schmitt, Luca Winterkamp

21. Mai 2023

## Aufgabe 5.1

Der code sieht wie Python aus, wir nehmen aber an, dass hier Arrayinitialisierung mit 1 beginnt.

```
_____ Arrayanfang sortieren _____  
def beginSort(A,k):  
    n = len(A)  
  
    quickSelect(A,1,n,k)  
  
    quickSort(A,1,k-1)
```

**Korrektheit** Mithilfe von Quickselect suchen wir das  $k$ -te Element des Arrays, welches die obere Schranke unseres zu sortierenden Arrayanfangs markiert. Danach gilt für jedes  $i, j$  mit  $1 \leq j < k < i \leq n$ :

$$A[j] \leq A[k] \leq A[i]$$

Nun muss lediglich der Array von 1 bis  $k-1$  mit einem Algorithmus in  $O(n \log n)$  sortiert werden. Wir verwenden Quicksort mit zufälliger Pivot-wahl, da die erwartete Laufzeit des Algorithmus  $O(n \log n)$  ist. Nach der Ausführung von Quicksort gilt:

$$A[1] \leq \dots \leq A[k-1] \leq A[k] \quad \text{und} \quad A[k] \leq A[i] \quad \forall k < i \leq n$$

Dies ist die Bedingung aus der Aufgabenstellung.

**Laufzeit** QuickSelect läuft in  $O(n)$  und unsere Wahl von QuickSort läuft erwartet in  $O(n \log n)$ . Da jedoch nur von 1 bis  $k-1$  sortiert wurde ist die Laufzeit in  $O(k \log k)$ . Die Laufzeiten zusammen:

$$O(n) + O(k \log k) = O(n + k \log k)$$

Die Laufzeitbedingung ist somit erfüllt.

## **Aufgabe 5.2**

## **Aufgabe 5.3**

## **Aufgabe 5.4**